



# Windows Sockets Error Codes

Socket Options

Winsock IOCTls

Winsock Annexes

Winsock Enumerations

Winsock Functions

Winsock Structures and Typedefs

Winsock Tracing Events

Winsock SPI

Windows Sockets Error Codes

178 out of 344 rated this helpful - [Rate this topic](#)**Applies to:** desktop apps only

Most Windows Sockets 2 functions do not return the specific cause of an error when the function returns. For information, see the [Handling Winsock Errors](#) topic.

The [WSAGetLastError](#) function returns the last error that occurred for the calling thread. When a particular Windows Sockets function indicates an error has occurred, this function should be called immediately to retrieve the extended error code for the failing function call. These error codes and a short text description associated with an error code are defined in the *Winerror.h* header file. The [FormatMessage](#) function can be used to obtain the message string for the returned error.

For information on how to handle error codes when porting socket applications to Winsock, see [Error Codes - errno, h\\_errno and WSAGetLastError](#).

The following list describes the possible error codes returned by the [WSAGetLastError](#) function. Errors are listed in numerical order with the error macro name. Some error codes defined in the *Winsock2.h* header file are not returned from any function.

Return code/value	Description
<b>WSA_INVALID_HANDLE</b> 6	Specified event object handle is invalid. An application attempts to use an event object, but the specified handle is not valid. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSA_NOT_ENOUGH_MEMORY</b> 8	Insufficient memory available. An application used a Windows Sockets function that directly maps to a Windows function. The Windows function is indicating a lack of required memory resources. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSA_INVALID_PARAMETER</b> 87	One or more parameters are invalid. An application used a Windows Sockets function which directly maps to a Windows function. The Windows function is indicating a problem with one or more parameters. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSA_OPERATION_ABORTED</b> 995	Overlapped operation aborted. An overlapped operation was canceled due to the closure of the socket, or the execution of the SIO_FLUSH command in <a href="#">WSAIoctl</a> . Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSA_IO_INCOMPLETE</b> 996	Overlapped I/O event object not in signaled state. The application has tried to determine the status of an overlapped operation which is not yet completed. Applications that use <a href="#">WSAGetOverlappedResult</a> (with the <i>fWait</i> flag set to <b>FALSE</b> ) in a polling mode to determine when an overlapped operation has completed, get this error code until the operation is complete. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSA_IO_PENDING</b> 997	Overlapped operations will complete later. The application has initiated an overlapped operation that cannot be completed immediately. A completion indication will be given later when the operation has been completed. Note that this error is returned by the operating system, so the error number may change in future releases of Windows.
<b>WSAEINTR</b>	Interrupted function call.

10004	A blocking operation was interrupted by a call to <a href="#">WSACancelBlockingCall</a> .
<b>WSAEBADF</b> 10009	File handle is not valid. The file handle supplied is not valid.
<b>WSAEACCES</b> 10013	Permission denied. An attempt was made to access a socket in a way forbidden by its access permissions. An example is using a broadcast address for <a href="#">sendto</a> without broadcast permission being set using <a href="#">setsockopt(SO_BROADCAST)</a> . Another possible reason for the WSAEACCES error is that when the <a href="#">bind</a> function is called (on Windows NT 4.0 with SP4 and later), another application, service, or kernel mode driver is bound to the same address with exclusive access. Such exclusive access is a new feature of Windows NT 4.0 with SP4 and later, and is implemented by using the <a href="#">SO_EXCLUSIVEADDRUSE</a> option.
<b>WSAEFAULT</b> 10014	Bad address. The system detected an invalid pointer address in attempting to use a pointer argument of a call. This error occurs if an application passes an invalid pointer value, or if the length of the buffer is too small. For instance, if the length of an argument, which is a <a href="#">sockaddr</a> structure, is smaller than the <code>sizeof(sockaddr)</code> .
<b>WSAEINVAL</b> 10022	Invalid argument. Some invalid argument was supplied (for example, specifying an invalid level to the <a href="#">setsockopt</a> function). In some instances, it also refers to the current state of the socket—for instance, calling <a href="#">accept</a> on a socket that is not listening.
<b>WSAEMFILE</b> 10024	Too many open files. Too many open sockets. Each implementation may have a maximum number of socket handles available, either globally, per process, or per thread.
<b>WSAEWOULDBLOCK</b> 10035	Resource temporarily unavailable. This error is returned from operations on nonblocking sockets that cannot be completed immediately, for example <a href="#">recv</a> when no data is queued to be read from the socket. It is a nonfatal error, and the operation should be retried later. It is normal for WSAEWOLDBLOCK to be reported as the result from calling <a href="#">connect</a> on a nonblocking <code>SOCK_STREAM</code> socket, since some time must elapse for the connection to be established.
<b>WSAEINPROGRESS</b> 10036	Operation now in progress. A blocking operation is currently executing. Windows Sockets only allows a single blocking operation—per task or thread—to be outstanding, and if any other function call is made (whether or not it references that or any other socket) the function fails with the WSAEINPROGRESS error.
<b>WSAEALREADY</b> 10037	Operation already in progress. An operation was attempted on a nonblocking socket with an operation already in progress—that is, calling <a href="#">connect</a> a second time on a nonblocking socket that is already connecting, or canceling an asynchronous request ( <a href="#">WSAAsyncGetXbyY</a> ) that has already been canceled or completed.
<b>WSAENOTSOCK</b> 10038	Socket operation on nonsocket. An operation was attempted on something that is not a socket. Either the socket handle parameter did not reference a valid socket, or for <a href="#">select</a> , a member of an <code>fd_set</code> was not valid.
<b>WSAEDESTADDRREQ</b> 10039	Destination address required. A required address was omitted from an operation on a socket. For example, this error is returned if <a href="#">sendto</a> is called with the remote address of <code>ADDR_ANY</code> .
<b>WSAEMSGSIZE</b> 10040	Message too long. A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram was smaller than the datagram itself.

<b>WSAEPROTOTYPE</b> 10041	<p>Protocol wrong type for socket.</p> <p>A protocol was specified in the <b>socket</b> function call that does not support the semantics of the socket type requested. For example, the ARPA Internet UDP protocol cannot be specified with a socket type of SOCK_STREAM.</p>
<b>WSAENOPROTOOPT</b> 10042	<p>Bad protocol option.</p> <p>An unknown, invalid or unsupported option or level was specified in a <b>getsockopt</b> or <b>setsockopt</b> call.</p>
<b>WSAEPROTONOSUPPORT</b> 10043	<p>Protocol not supported.</p> <p>The requested protocol has not been configured into the system, or no implementation for it exists. For example, a <b>socket</b> call requests a SOCK_DGRAM socket, but specifies a stream protocol.</p>
<b>WSAESOCKTNOSUPPORT</b> 10044	<p>Socket type not supported.</p> <p>The support for the specified socket type does not exist in this address family. For example, the optional type SOCK_RAW might be selected in a <b>socket</b> call, and the implementation does not support SOCK_RAW sockets at all.</p>
<b>WSAEOPNOTSUPP</b> 10045	<p>Operation not supported.</p> <p>The attempted operation is not supported for the type of object referenced. Usually this occurs when a socket descriptor to a socket that cannot support this operation is trying to accept a connection on a datagram socket.</p>
<b>WSAEPFNOSUPPORT</b> 10046	<p>Protocol family not supported.</p> <p>The protocol family has not been configured into the system or no implementation for it exists. This message has a slightly different meaning from WSAEAFNOSUPPORT. However, it is interchangeable in most cases, and all Windows Sockets functions that return one of these messages also specify WSAEAFNOSUPPORT.</p>
<b>WSAEAFNOSUPPORT</b> 10047	<p>Address family not supported by protocol family.</p> <p>An address incompatible with the requested protocol was used. All sockets are created with an associated address family (that is, AF_INET for Internet Protocols) and a generic protocol type (that is, SOCK_STREAM). This error is returned if an incorrect protocol is explicitly requested in the <b>socket</b> call, or if an address of the wrong family is used for a socket, for example, in <b>sendto</b>.</p>
<b>WSAEADDRINUSE</b> 10048	<p>Address already in use.</p> <p>Typically, only one usage of each socket address (protocol/IP address/port) is permitted. This error occurs if an application attempts to <b>bind</b> a socket to an IP address/port that has already been used for an existing socket, or a socket that was not closed properly, or one that is still in the process of closing. For server applications that need to <b>bind</b> multiple sockets to the same port number, consider using <b>setsockopt</b> (SO_REUSEADDR). Client applications usually need not call <b>bind</b> at all—<b>connect</b> chooses an unused port automatically. When <b>bind</b> is called with a wildcard address (involving ADDR_ANY), a WSAEADDRINUSE error could be delayed until the specific address is committed. This could happen with a call to another function later, including <b>connect</b>, <b>listen</b>, <b>WSAConnect</b>, or <b>WSAJoinLeaf</b>.</p>
<b>WSAEADDRNOTAVAIL</b> 10049	<p>Cannot assign requested address.</p> <p>The requested address is not valid in its context. This normally results from an attempt to <b>bind</b> to an address that is not valid for the local computer. This can also result from <b>connect</b>, <b>sendto</b>, <b>WSAConnect</b>, <b>WSAJoinLeaf</b>, or <b>WSASendTo</b> when the remote address or port is not valid for a remote computer (for example, address or port 0).</p>
<b>WSAENETDOWN</b> 10050	<p>Network is down.</p> <p>A socket operation encountered a dead network. This could indicate a serious failure of the network system (that is, the protocol stack that the Windows Sockets DLL runs over), the network interface, or the local network itself.</p>
<b>WSAENETUNREACH</b> 10051	<p>Network is unreachable.</p> <p>A socket operation was attempted to an unreachable network. This usually means the local software knows no route to reach the remote host.</p>

<b>WSAENETRESET</b> 10052	Network dropped connection on reset. The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress. It can also be returned by <b>setsockopt</b> if an attempt is made to set <b>SO_KEEPALIVE</b> on a connection that has already failed.
<b>WSAECONNABORTED</b> 10053	Software caused connection abort. An established connection was aborted by the software in your host computer, possibly due to a data transmission time-out or protocol error.
<b>WSAECONNRESET</b> 10054	Connection reset by peer. An existing connection was forcibly closed by the remote host. This normally results if the peer application on the remote host is suddenly stopped, the host is rebooted, the host or remote network interface is disabled, or the remote host uses a hard close (see <b>setsockopt</b> for more information on the <b>SO_LINGER</b> option on the remote socket). This error may also result if a connection was broken due to keep-alive activity detecting a failure while one or more operations are in progress. Operations that were in progress fail with <b>WSAENETRESET</b> . Subsequent operations fail with <b>WSAECONNRESET</b> .
<b>WSAENOBUFS</b> 10055	No buffer space available. An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.
<b>WSAEISCONN</b> 10056	Socket is already connected. A connect request was made on an already-connected socket. Some implementations also return this error if <b>sendto</b> is called on a connected <b>SOCK_DGRAM</b> socket (for <b>SOCK_STREAM</b> sockets, the <b>to</b> parameter in <b>sendto</b> is ignored) although other implementations treat this as a legal occurrence.
<b>WSAENOTCONN</b> 10057	Socket is not connected. A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using <b>sendto</b> ) no address was supplied. Any other type of operation might also return this error—for example, <b>setsockopt</b> setting <b>SO_KEEPALIVE</b> if the connection has been reset.
<b>WSAESHUTDOWN</b> 10058	Cannot send after socket shutdown. A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous <b>shutdown</b> call. By calling <b>shutdown</b> a partial close of a socket is requested, which is a signal that sending or receiving, or both have been discontinued.
<b>WSAETOOMANYREFS</b> 10059	Too many references. Too many references to some kernel object.
<b>WSAETIMEDOUT</b> 10060	Connection timed out. A connection attempt failed because the connected party did not properly respond after a period of time, or the established connection failed because the connected host has failed to respond.
<b>WSAECONNREFUSED</b> 10061	Connection refused. No connection could be made because the target computer actively refused it. This usually results from trying to connect to a service that is inactive on the foreign host—that is, one with no server application running.
<b>WSAELOOP</b> 10062	Cannot translate name. Cannot translate a name.
<b>WSAENAMETOOLONG</b> 10063	Name too long. A name component or a name was too long.
<b>WSAEHOSTDOWN</b> 10064	Host is down. A socket operation failed because the destination host is down. A socket operation

encountered a dead host. Networking activity on the local host has not been initiated. These conditions are more likely to be indicated by the error WSAETIMEDOUT.

**WSAEHOSTUNREACH**  
10065

No route to host.  
A socket operation was attempted to an unreachable host. See WSAENETUNREACH.

**WSAENOTEMPTY**  
10066

Directory not empty.  
Cannot remove a directory that is not empty.

**WSAEPROCLIM**  
10067

Too many processes.  
A Windows Sockets implementation may have a limit on the number of applications that can use it simultaneously. [WSAStartup](#) may fail with this error if the limit has been reached.

**WSAEUSERS**  
10068

User quota exceeded.  
Ran out of user quota.

**WSAEDQUOT**  
10069

Disk quota exceeded.  
Ran out of disk quota.

**WSAESTALE**  
10070

Stale file handle reference.  
The file handle reference is no longer available.

**WSAEREMOTE**  
10071

Item is remote.  
The item is not available locally.

**WSASYSNOTREADY**  
10091

Network subsystem is unavailable.  
This error is returned by [WSAStartup](#) if the Windows Sockets implementation cannot function at this time because the underlying system it uses to provide network services is currently unavailable. Users should check:

- That the appropriate Windows Sockets DLL file is in the current path.
- That they are not trying to use more than one Windows Sockets implementation simultaneously. If there is more than one Winsock DLL on your system, be sure the first one in the path is appropriate for the network subsystem currently loaded.
- The Windows Sockets implementation documentation to be sure all necessary components are currently installed and configured correctly.

**WSAVERNOTSUPPORTED**  
10092

Winsock.dll version out of range.  
The current Windows Sockets implementation does not support the Windows Sockets specification version requested by the application. Check that no old Windows Sockets DLL files are being accessed.

**WSANOTINITIALISED**  
10093

Successful WSAStartup not yet performed.  
Either the application has not called [WSAStartup](#) or [WSAStartup](#) failed. The application may be accessing a socket that the current active task does not own (that is, trying to share a socket between tasks), or [WSACleanup](#) has been called too many times.

**WSAEDISCON**  
10101

Graceful shutdown in progress.  
Returned by [WSARcvFrom](#) and [WSASendTo](#) to indicate that the remote party has initiated a graceful shutdown sequence.

**WSAENOMORE**  
10102

No more results.  
No more results can be returned by the [WSALookupServiceNext](#) function.

**WSAECANCELLED**  
10103

Call has been canceled.  
A call to the [WSALookupServiceEnd](#) function was made while this call was still processing. The call has been canceled.

<b>WSAEINVALIDPROCTABLE</b> 10104	Procedure call table is invalid. The service provider procedure call table is invalid. A service provider returned a bogus procedure table to Ws2_32.dll. This is usually caused by one or more of the function pointers being <b>NULL</b> .
<b>WSAEINVALIDPROVIDER</b> 10105	Service provider is invalid. The requested service provider is invalid. This error is returned by the <b>WSCGetProviderInfo</b> and <b>WSCGetProviderInfo32</b> functions if the protocol entry specified could not be found. This error is also returned if the service provider returned a version number other than 2.0.
<b>WSAEPROVIDERFAILEDINIT</b> 10106	Service provider failed to initialize. The requested service provider could not be loaded or initialized. This error is returned if either a service provider's DLL could not be loaded ( <b>LoadLibrary</b> failed) or the provider's <b>WSPStartup</b> or <b>NSPStartup</b> function failed.
<b>WSASYSALLFAILURE</b> 10107	System call failure. A system call that should never fail has failed. This is a generic error code, returned under various conditions. Returned when a system call that should never fail does fail. For example, if a call to <b>WaitForMultipleEvents</b> fails or one of the registry functions fails trying to manipulate the protocol/namespace catalogs. Returned when a provider does not return SUCCESS and does not provide an extended error code. Can indicate a service provider implementation error.
<b>WSASERVICE_NOT_FOUND</b> 10108	Service not found. No such service is known. The service cannot be found in the specified name space.
<b>WSATYPE_NOT_FOUND</b> 10109	Class type not found. The specified class was not found.
<b>WSA_E_NO_MORE</b> 10110	No more results. No more results can be returned by the <b>WSALookupServiceNext</b> function.
<b>WSA_E_CANCELLED</b> 10111	Call was canceled. A call to the <b>WSALookupServiceEnd</b> function was made while this call was still processing. The call has been canceled.
<b>WSAEREFUSED</b> 10112	Database query was refused. A database query failed because it was actively refused.
<b>WSAHOST_NOT_FOUND</b> 11001	Host not found. No such host is known. The name is not an official host name or alias, or it cannot be found in the database(s) being queried. This error may also be returned for protocol and service queries, and means that the specified name could not be found in the relevant database.
<b>WSATRY_AGAIN</b> 11002	Nonauthoritative host not found. This is usually a temporary error during host name resolution and means that the local server did not receive a response from an authoritative server. A retry at some time later may be successful.
<b>WSANO_RECOVERY</b> 11003	This is a nonrecoverable error. This indicates that some sort of nonrecoverable error occurred during a database lookup. This may be because the database files (for example, BSD-compatible HOSTS, SERVICES, or PROTOCOLS files) could not be found, or a DNS request was returned by the server with a severe error.
<b>WSANO_DATA</b> 11004	Valid name, no data record of requested type. The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for. The usual example for this is a host name-to-address translation attempt (using <b>gethostbyname</b> or <b>WSAAsyncGetHostByName</b> ) which uses the DNS (Domain Name Server). An MX record is returned but no A record—indicating the host

itself exists, but is not directly reachable.

<b>WSA_QOS_RECEIVERS</b> 11005	QoS receivers. At least one QoS reserve has arrived.
<b>WSA_QOS_SENDERS</b> 11006	QoS senders. At least one QoS send path has arrived.
<b>WSA_QOS_NO_SENDERS</b> 11007	No QoS senders. There are no QoS senders.
<b>WSA_QOS_NO_RECEIVERS</b> 11008	QoS no receivers. There are no QoS receivers.
<b>WSA_QOS_REQUEST_CONFIRMED</b> 11009	QoS request confirmed. The QoS reserve request has been confirmed.
<b>WSA_QOS_ADMISSION_FAILURE</b> 11010	QoS admission error. A QoS error occurred due to lack of resources.
<b>WSA_QOS_POLICY_FAILURE</b> 11011	QoS policy failure. The QoS request was rejected because the policy system couldn't allocate the requested resource within the existing policy.
<b>WSA_QOS_BAD_STYLE</b> 11012	QoS bad style. An unknown or conflicting QoS style was encountered.
<b>WSA_QOS_BAD_OBJECT</b> 11013	QoS bad object. A problem was encountered with some part of the filterspec or the provider-specific buffer in general.
<b>WSA_QOS_TRAFFIC_CTRL_ERROR</b> 11014	QoS traffic control error. An error with the underlying traffic control (TC) API as the generic QoS request was converted for local enforcement by the TC API. This could be due to an out of memory error or to an internal QoS provider error.
<b>WSA_QOS_GENERIC_ERROR</b> 11015	QoS generic error. A general QoS error.
<b>WSA_QOS_ESERVICETYPE</b> 11016	QoS service type error. An invalid or unrecognized service type was found in the QoS flowspec.
<b>WSA_QOS_EFLOWSPEC</b> 11017	QoS flowspec error. An invalid or inconsistent flowspec was found in the <b>QOS</b> structure.
<b>WSA_QOS_EPROVSPECBUF</b> 11018	Invalid QoS provider buffer. An invalid QoS provider-specific buffer.
<b>WSA_QOS_EFILTERSTYLE</b> 11019	Invalid QoS filter style. An invalid QoS filter style was used.
<b>WSA_QOS_EFILTERTYPE</b> 11020	Invalid QoS filter type. An invalid QoS filter type was used.
<b>WSA_QOS_EFILTERCOUNT</b>	Incorrect QoS filter count.

11021	An incorrect number of QoS FILTERSPECS were specified in the FLOWDESCRIPTOR.
<b>WSA_QOS_EOBJLENGTH</b> 11022	Invalid QoS object length. An object with an invalid ObjectLength field was specified in the QoS provider-specific buffer.
<b>WSA_QOS_EFLOWCOUNT</b> 11023	Incorrect QoS flow count. An incorrect number of flow descriptors was specified in the QoS structure.
<b>WSA_QOS_EUNKOWNPSOBJ</b> 11024	Unrecognized QoS object. An unrecognized object was found in the QoS provider-specific buffer.
<b>WSA_QOS_EPOLICYOBJ</b> 11025	Invalid QoS policy object. An invalid policy object was found in the QoS provider-specific buffer.
<b>WSA_QOS_EFLOWDESC</b> 11026	Invalid QoS flow descriptor. An invalid QoS flow descriptor was found in the flow descriptor list.
<b>WSA_QOS_EPSFLOWSPEC</b> 11027	Invalid QoS provider-specific flowspec. An invalid or inconsistent flowspec was found in the QoS provider-specific buffer.
<b>WSA_QOS_EPSFILTERSPEC</b> 11028	Invalid QoS provider-specific filterspec. An invalid FILTERSPEC was found in the QoS provider-specific buffer.
<b>WSA_QOS_ESDMODEOBJ</b> 11029	Invalid QoS shape discard mode object. An invalid shape discard mode object was found in the QoS provider-specific buffer.
<b>WSA_QOS_ESHAPERATEOBJ</b> 11030	Invalid QoS shaping rate object. An invalid shaping rate object was found in the QoS provider-specific buffer.
<b>WSA_QOS_RESERVED_PETYPE</b> 11031	Reserved policy QoS element type. A reserved policy element was found in the QoS provider-specific buffer.

## Requirements

**Header** Winsock2.h;  
Winerror.h

## See also

[Error Codes - errno, h\\_errno and WSAGetLastError](#)  
[Handling Winsock Errors](#)  
[FormatMessage](#)  
[WSAGetLastError](#)

[Send comments about this topic to Microsoft](#)

Build date: 5/5/2012

Did you find this helpful?  Yes  No

## Community Additions

ADD

java.net.SocketException: Unrecognized Windows Sockets error: 126: create

**java.net.SocketException: Unrecognized Windows Sockets error: 126: create \$0\$0 \$0 \$0Any clues? Can't find any references for this code.\$0**



### Socket Error

Hi,

I am trying to get response from <http://obix.tridium.com/obix> thru "ReQuery" method. I am getting "Socket Error #10057(sometimes 10061) Connection refused". Please help me How can I resolve this issue.

Codebase is in Delphi.

Thanks



hello

hey i dont know what is wrong



### Network Link Error (-1)

Dear Team,

We are running a Kerridge system(Auto Line) on our 2003 Standard Edition R2 OS. We have been receiving an error called "Network link disconnected from the host end (-1)" from client -KCML Client..We could access files and folders on the server. The KCML server is connected to the domain.

Prasath.K  
+967 1 22 77 854



### Network sites

msdn

### Windows Dev Center

Windows Dev Center home  
Metro style apps  
Internet Explorer  
Desktop  
Hardware

### Other links

Support  
Microsoft Connect

Microsoft

© 2012 Microsoft. All rights reserved.

[Terms of use](#) | [Trademarks](#) | [Privacy statement](#) | [Site feedback](#) | [United States \(English\)](#)